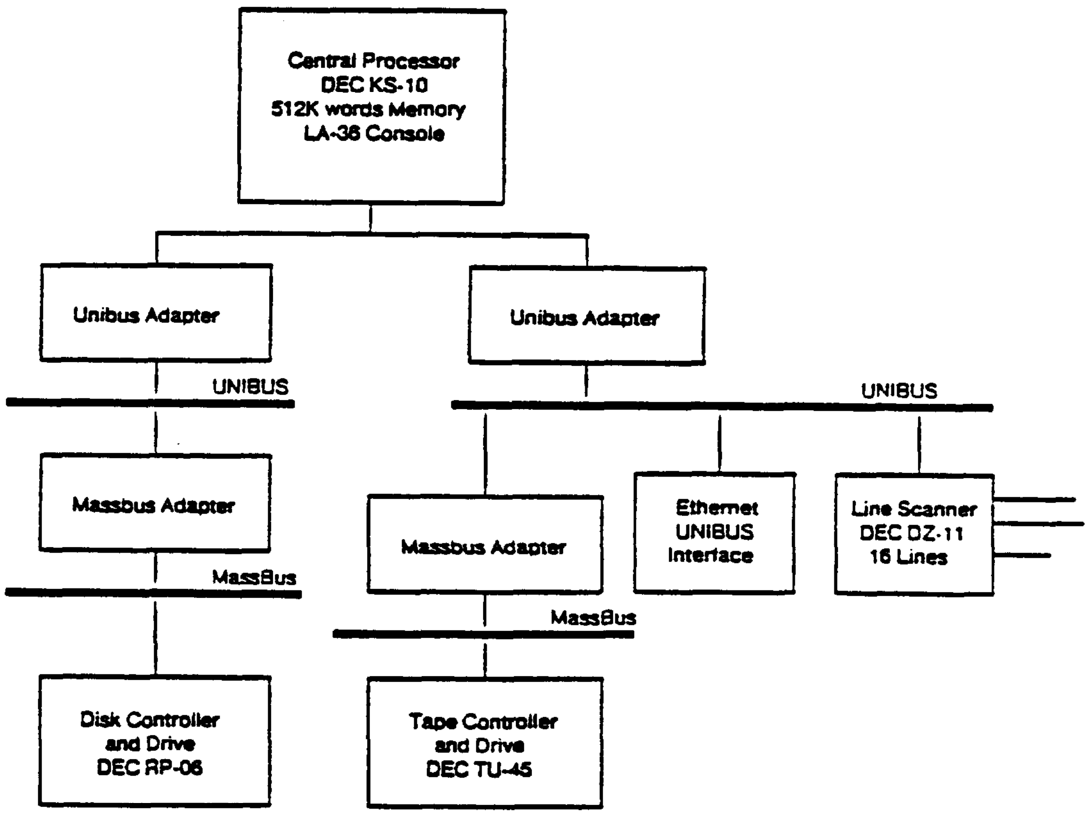

Figure 2: SUMEX-AIM DEC 2020 Configuration

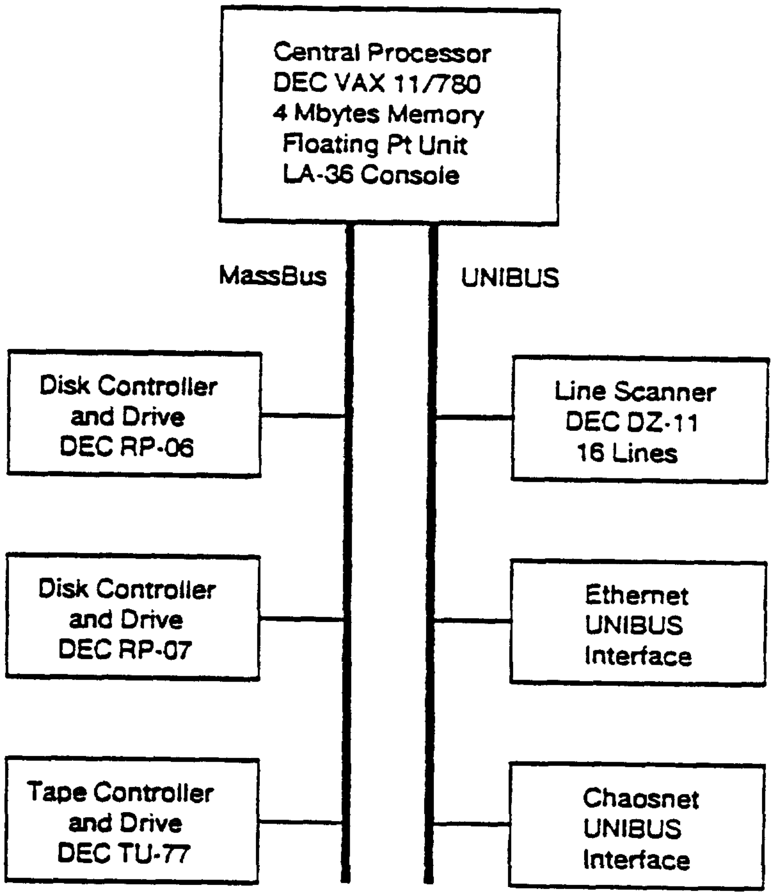

Figure 3: SUMEX-AIM Shared DEC VAX 11/780 Configuration

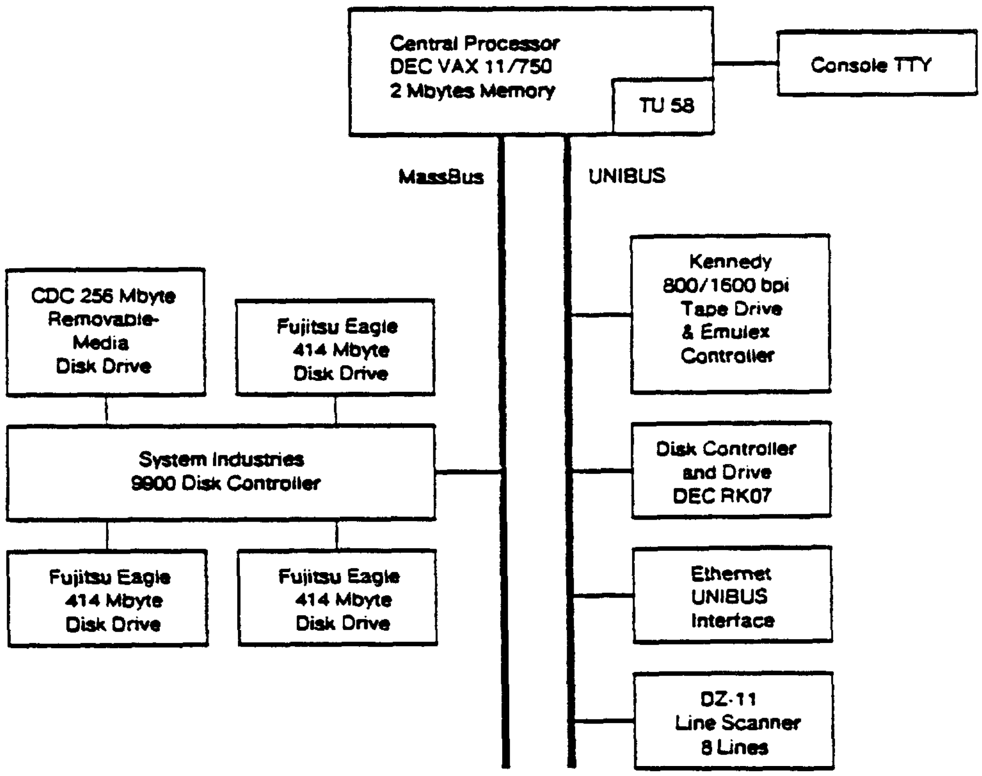

Figure 4: SUMEX-AIM File Server Configuration

III.A.3.3. Core System Development

Operating System Software

The various hardware elements of the SUMEX-AIM computing environment require the development and support of the operating systems that provide the interface between user software and the raw computing capacity. In addition to performance and relevance to AI research, much of our strategy for hardware selection has been based on being able to share development of the operating systems for our research among a large computer science community. This includes the mainframe systems (TOPS-20 and UNIX) and the workstation systems. Following are some highlights of recent system software developments.

TOPS-20 Development

The upgrade of the KI-TENEX system to the 2060 required a very large effort. Whereas the KI-TENEX system contained a great many local enhancements and adaptations, our goal was to run a TOPS-20 system that was broadly supported but which also tracked research developments outside of those motivated by vendor commercial interests. The most obvious choice for our immediate system peer community was the other 6 DEC 2060 sites at Stanford since we shared common internet problems and also had common goals in supporting research work rather than production computing. We also, of course, retained contact with the other ARPANET computer science systems. This course has constrained our own local developments by being part of a larger group of peers but the added problems of coordination have required fewer site-specific extensions and customizations at the operating system level.

Given this perspective, the following are specific areas of TOPS-20 system effort:

- In the conversion from TENEX, much planning and effort went into moving the file system, along with the pertinent user-specific directory information. In addition, we were able to preserve access to the vast magnetic tape library of archived and otherwise backed up files that had been created and saved since the inception of SUMEX. A TOPS-20 version of BSYS, a file archiving system, was imported from ISI as part of the effort to convert to the 2060. Numerous changes were made to make it compatible with the version of BSYS previously used at SUMEX. The LOOKUP program, used under TENEX, was converted to TOPS-20 use and made compatible with the new version of BSYS. We reviewed and updated appropriate documentation files in the HLP: and DOC: directories. And we identified and upgraded numerous system utility programs that utilized TENEX-dependent system calls.

- Using Tenex code previously developed at SUMEX as a base, we added new code to the TOPS-20 monitor to significantly enhance the user interface to the file system naming primitives. One addition was intercepting a ? typed by a user as part of a file name, then displaying for the user the valid file name alternatives matching the type-in up to that point, and finally returning to the original context, allowing the user to continue typing where he left off. Another addition was to generalize the logic involved in file name recognition in the case where more than one file matches what is typed in at the point where the request for recognition was given. The new logic looks ahead at the alternatives and fills out as much of the file name as possible, i.e. up to the point of ambiguity.

- Continued development of QANAL (formerly ANAL), a crash analysis program that has been under development since 1978. This program significantly eases the burden of analyzing the causes of system crashes due to both hardware and software problems. In addition, the accumulated outputs from QANAL allow for the detection of long term crash correlations to analyze infrequent problems.

- Track network protocol and service (e.g., file transfer and electronic mail) developments. We coordinated SUMEX's changes required to support the ARPANET-wide change from the old NCP protocols to the DOD IP/TCP protocols. This complex software required significant effort on our part because SUMEX-AIM has become a major communications crossroads and so exercises the network code very heavily. This has raised many problems of bugs and performance that we have worked to improve. We have played an active role in network discussion groups related to areas such as electronic mail, network designs, and protocols and had kept system tables for network host names and addresses, both local and over the ARPANET, up-to-date.

- Developed expanded file system support through multiple RP07 disk drive service. We were the first site to support more than one RP07 unit in a single structure.

- Implemented support for the old but superior LP10 printer from the KI-TENEX system. Even though DEC doesn't support this configuration, the LP10 has become our standard printer.

- Implemented subdirectory access to allow users full "owner" access to their subdirectories via the Access Control Job.

- Developed improved system allocation code, including the ability to withhold scheduler "windfall" from a given class or classes, with associated code in SKED% JSYS.

- Improved the efficiency of file backup and archive facilities by flagging directories with ARCHIVE and MIGRATE requests pending rather than searching through all directories serially.

- We have done substantial work on the TOPS-20 system Executive, the program that serves as the primary interface between users and the system. It provides commands to manipulate files, directories, and devices; control job and terminal parameter settings; observe job and system status; and execute public and private programs. The SUMEX EXEC is quite well developed at this stage but we have made several improvements. For example, we added a command line editor developed at the University of Texas and commands for the various laser printer spooling capabilities described later. There were also many more minor upgrades such as reading SYSTEM:LOGIN.CMD and SYSTEM:COMAND.CMD files on user login, account verification, enhancing various information commands, and improved directory and file system facilities to assist users in managing their files.

We have made numerous monitor bug and hardware problem repairs to provide for more reliable system operation and file integrity. Obvious bugs were removed long ago so those remaining are elusive and difficult to track down. We have also spent time keeping up-to-date with the latest monitor releases.

VAX 4.2 BSD UNIX Development

We run UNIX on our shared VAX 11/780 and on our 11/750 file servers. This system has been used pretty much as distributed by the University of California at Berkeley, except for local network support modifications. The local VAX user community is small so we have not expended much system effort beyond staying current with operating system releases and with useful UNIX community developments. The SUMEX VAX was the first site at Stanford to bring up the Berkeley 4.2 BSD distribution in October 1983. Since this was an early distribution, there were quite a number of bug fixes required; these were accomplished both through local effort and through monitoring the unix-wizards mailing list. After this kernel was running on the SUMEX machine, it was transported other sites and became the basis for the campus-wide UNIX 4.2 distribution.

To allow the UNIX network interface code to work in our Stanford subnet environment, we created a pseudo-network interface driver called 'sub0', that routed all output IP datagrams, based on their subnet numbers. This driver was done transparently, so that at system boot time, you could configure the machine for Stanford subnets, or for normal network routing. We also worked with other Stanford sites to install the Stanford PUP network drivers and servers back into 4.2 BSD (Berkeley does not support these).

Workstation System Development

Lisp workstations represent the major new direction for system development at SUMEX-AIM because these machines offer high performance Lisp engines, large address spaces required for sophisticated AI systems, flexible graphics interfaces for users, state-of-the-art program development and debugging tools, and a modularity that promises to be the vehicle for disseminating AI systems into user environments. We have accordingly invested a large part of our system effort in developing selected workstations and the related networking environments for effective use in the SUMEX-AIM community.

Xerox D-Machines

Much of the SUMEX-AIM community uses InterLisp and has moved naturally to the Xerox D-machines -- initially the Dolphin and then the Dandelion, Dandetiger, and Dorado. Much work has gone into hardware installation and networking support but we have also developed numerous software packages to help make the machines more effective for users and to ease our own problems in managing the distributed workstation environment.

In the transition to workstations as *computing* environments suitable for AI applications work, not just as programming environments, much system development remains to be done. One of the problems we have examined and plan to continue to exploring is that of building distributed expert systems. We are interested, for example, in separating the reasoning components and user interfaces and are designing a system with multiple processes which can run on a single or multiple workstations in order to independently develop, tune and evaluate the components. To facilitate this we have developed a prototype inter-process message passing interface which makes the topology of the system invisible to communicating processes, whether on one machine or several CPU's linked via the Ethernet.

Another of our interests is in exploring how to combine different software and/or hardware architectures in order to take advantage of the best features of each. One simple low level program that we built allows us to use Interlisp workstations to down load software into Mesa workstations in order to boot them using the Ethernet as an alternative to the hard or floppy disk drives. Along the same lines, we are exploring efficient ways to communicate high level descriptions of graphic data among differing media. We have developed a simple system which will take text formatting files and translate them into graphic window displays, defining active regions of the screen in the process. This facilitates the design of user interfaces using the familiar medium of text processing.

In our AI systems work, we have developed a low overhead object-oriented system which is designed to be flexible enough to model different object-oriented programming styles at the same time. It is also designed to facilitate a model of large knowledge bases which reside principally on file servers but whose components are loaded on demand. With this system, a minimal set of information about all the objects in a knowledge base is loaded upon opening. This information allows many simple inquires about the nature of objects and their relationships to be made without the main body of the object being resident. Only when non-trivial operations are performed are the contents of the object brought into core. This design is based on the belief that the size of knowledge bases will eventually grow to exceed the capacity of any given computer. However, most systems will generally only need a manageable subset of objects at runtime.

Other work we have done includes monitoring tools to examine static function calling hierarchy as well as view runtime executions graphically. We are also developing graphics interfaces to knowledge base construction and maintenance.

Some of the InterLisp software packages that have been written in the course of this work include:

ACFontCreate -- Reads a Xerox PARC font file in AC format into a lisp data structure

BaudRate -- Benchmarks baudrates by BINing through a file

DSys -- Monitors D machine usage on demand

GraphNet -- Derives topology of the PUP internet via net and gateway probes

HPColor -- Interlisp image stream implementation to drive H-P dgl graphics

Impress -- Interlisp image stream implementation to generate Impress print files

MakeStrike -- Writes out an Interlisp display font as a strike file

MLabel -- Generates mailing labels from a mailing list

RasterFontCreate -- Generates an Impress font of bitmap patches in arbitrary scale

ReadRSTFontFile -- Reads an Impress font file into a list data structure

RemoteTools -- Tools to manipulate a remote Interlisp using its systat process

RootPicture -- Reads a Press file bitmap into a lisp bitmap

RSTSample -- Creates an Impress sampler showing all characters of a font

SIL -- Reads and displays a SIL drawing file and optionally hardcopies it

SYSTAT -- a remote Eval server for Interlisp

Undither -- Compresses a previously dithered image into an AIS file

VDSDog -- Monitors array space usage to prevent crashing from lack thereof

WriteRSTFontFile -- generates an Impress font file from a special Lisp structure

ZDir -- TENEX-style directory lister for use with UNIX via Leaf server calls

DScribe -- A simple SCRIBE-to-display list parser/driver.

EtherBoot -- Provides microcode and program boot service for Xerox 8000's

GraphCalls -- Graphs the calling hierarchy of a lisp function and more

Hash -- Provide a machine independent hash file facility

EditBG -- A background/border texture editor.

FileLstW -- Menu-based interface to the file package.

MagnifyW -- A magnifying glass for bitmaps.

Message -- Multi-process/Multi-CPU message passing facility.

MultiW -- Links windows so that they move, surface, and close as a group

OZone -- An object-oriented programming system for Interlisp

Plotter -- Interlisp image stream to generate native-mode H-P plot files

Register -- Bundles menus into a coherent device for complex input

Region -- A utility to allow dissimilar activity in a single window.

Storage -- A utility to display Interlisp data type storage graphically.

Once a package has been developed and determined to be of general interest, we announce it over an electronic mail users list and make it available to other sites. In some cases, packages have such extensive utility that they are submitted as LispUsers packages for distribution by Xerox. This occurred in the case of Graphcalls, Hash, MultiW, and FileLstW, the latter submitted under the name Manager.

We have worked closely with many other sites, including the Center for Study of Language and Information at Stanford, the Stanford Campus Networking group, Rutgers University, Ohio State University, the University of Pittsburgh, Cornell, Maryland, and industrial research groups such as Xerox Palo Alto Research Center, SRI, Teknowledge, IntelliCorp, and Schlumberger-Doll Research. We have been the maintainers for the international electronic mail network of users for research D-machines, which have upwards of 300 readers, and the interchange of ideas and problems among this group has been of great service to all users.

Symbolics Lisp Machines

We have a growing community of Symbolics machines and users. Little development has gone into the tools for these systems yet because the small number of machines we have are concentrated in applications groups. We have actively supported the installation and maintenance of these systems, the installation of new software releases, and the integration of these systems with the rest of our networking environment. We were a beta test site for the Symbolics IP/TCP software.

Macintosh Workstations

In early 1984 Apple Computer released their new Macintosh and we were immediately interested in it as a possible low-cost display workstation to interface to our Lisp workstations and other hosts. In order to evaluate the Macintosh for this purpose, SUMEX received some early equipment and manuals through Stanford's participation in the Apple university consortium program. Like many groups trying to experiment with Macintosh software however, we found the Apple Lisa cross-development environment somewhat restrictive and hard to use and this was the only way to create Macintosh software at the time. So we built a UNIX-based cross-development environment on our VAX. It turns out, that this was the first C development environment available on the Macintosh when we released our software (via Arpanet FTP) in June of 1984. SUMacC (Stanford University Macintosh C) has been quite widely received, and is in use at well over a hundred sites throughout the US and in foreign countries. SUMACC integrated pieces of software from many groups, and was therefore something of a cooperative effort. We have openly distributed it to other users either through network FTP or a magnetic tape at distribution cost. Version 2.0 of the SUMACC system was released in November of 1984.

Among the many useful programs subsequently written with SUMACC were: (1) a Kermit program done at Harvard, (2) the Mac PSL (Portable Standard LISP) done at the University of Utah, and (3) an 'external file system' done by John Seamons of LucasFilm which allows the Macintosh to use an Ethernet host (such as UNIX) as a general network file server (see also page 37).

With the increased usage of Macintoshes in the SUMEX-AIM community, the need to be able to transfer files between them and TOPS-20 mainframes quickly arose. We therefore reimplemented the MACGet and MACPut file transfer utilities, previously developed for UNIX, for TOPS-20. These incorporated TOPS-20 style terminal handling and file system conventions. Both programs provide reliable (i.e., checksummed) transfer of either text or binary data, and are now gaining wide-spread use outside of SUMEX.

Virtual Workstation Graphics

Finally, we have done a number of experiments with the remote connection of bitmapped displays to hosts and workstations. Generally, the displays on Lisp machines are tethered through a high bandwidth cable to their processors. This limits the flexibility with which users can move from one Lisp machine to another (one must move physically to another machine) and loses the ability of researchers to work from home over telephone lines. A way of providing more flexible display to processor connection is to use a virtual graphics protocol, such as the V Kernel system developed by Lantz [18], that allows efficient communication of the contents to be displayed on a bitmapped screen. In an initial experiment, an Interlisp virtual graphics module was written to run on the DEC-2060 and drive the graphics engine of a Sun MicroSystems workstation over the Ethernet. This system allows applications running on the DEC-2060 to create views, and windows within those views on the remote workstation, and then using the Virtual Graphics Terminal Protocols, manipulate those views and windows. One can place text, draw objects such as points, lines, shaded rectangles, splines, and bitmaps in these screen areas. Local and remote editing of the graphics representation is also possible with a responsiveness close to that of a directly connected display.

Network Services

A highly important aspect of the SUMEX system is effective communication within our growing distributed computing environment and with remote users. In addition to the economic arguments for terminal access, networking offers other advantages for shared computing. These include improved inter-user communications, more effective software sharing, uniform user access to multiple machines and special purpose resources, convenient file transfers, more effective backup, and co-processing between remote machines. Networks are crucial for maintaining the collaborative scientific and software contacts within the SUMEX-AIM community.

Remote Networks

We continue our connection to TYMNET as the primary means for access to SUMEX-AIM from research groups around the country and abroad. Substantial work was required to transfer TYMNET service from the KI-TENEX system to the 2060 because the new system does not support the same memory-sharing interface we had for the KI-10's. There has been no significant change in user service or network performance though. Very limited facilities for file transfer exist and no improvements appear to be forthcoming soon. Services continue to be purchased jointly with the Rutgers Computers in Biomedicine resource to maximize our volume usage price break. We continue to have serious difficulties getting needed service from TYMNET for debugging network problems and users away from major cities have problems with echo response times.

We also continue our extremely advantageous connection to the Department of Defense's ARPANET, managed by the Defense Communications Agency (DCA). This connection has been possible because of the long-standing basic research effort in AI within the Knowledge Systems Laboratory that is funded by DARPA. Terminal access restrictions are in force so that only users affiliated with DoD-supported contractors may use TELNET facilities. ARPANET is the primary link between SUMEX and other machine resources such as Rutgers-AIM and the large AI computer science community supported by DARPA. Our early Honeywell IMP has been upgraded to a BBN C/30 IMP in preparation for the transition to the IP/TCP protocols. We are also investigating the installation of a link to the DARPA wideband satellite network to facilitate the rapid transfer of large amounts of data such as are involved with projects like our Concurrent Symbolic Computing Architectures project.

Local Area Networks

For many years now, we have been developing our local area networking systems to enhance the facilities available to researchers. Much of this work has centered on the effective integration of distributed computing resources in the form of mainframes, workstations, and servers. Network gateways and terminal interface processors (TIP's) were developed and extended to link our environment together and are now the standard system used in the campus-wide Stanford University network. We are developing gateways to interface other equipment as needed too (e.g., the Macintosh and Lisa). A diagram of our local area network system is shown in Figure 5 on 36 and the following summarizes our LAN-related development work.

MC-68000 Server Kernel -- Our early network gateways and TIP's were based on PDP-11 systems. But these soon became limiting in terms of speed, address space, and cost. With the introduction of the Motorola MC-68000 microprocessor and its integration into a compact, large-memory machine in the prototype SUN processor board developed in the Computer Systems Laboratory at Stanford, a much better vehicle was at hand. The net server software we developed for the PDP-11 included a kernel which handles hardware interfaces, core allocation, process scheduling, and low-level network protocol management. The 3 MBit/sec Ethernet PDP-11 based PUP kernel was translated and augmented for the MC-68000 CPU/SUN ethernet interface. This kernel then became the basis for the SUMEX gateway and TIP software which both have become the Stanford standard. As networking technology developed, the SUMEX kernel was extended to include 10 MBit/sec Ethernet drivers and to support 10 Mbit/sec PUP, XNS, and IP protocols. The main modification needed was the addition of a 10 MBit/sec Ethernet address resolution protocol module so that a 10 MBit/sec PUP host could discover its "soft" PUP address from a cooperating gateway on its local network.

Ethernet TIP -- Based on the new augmented MC-68000 kernel, the 3 Mbit/sec PDP-11 Ether TIP code was translated. This new TIP could handle increments of 8

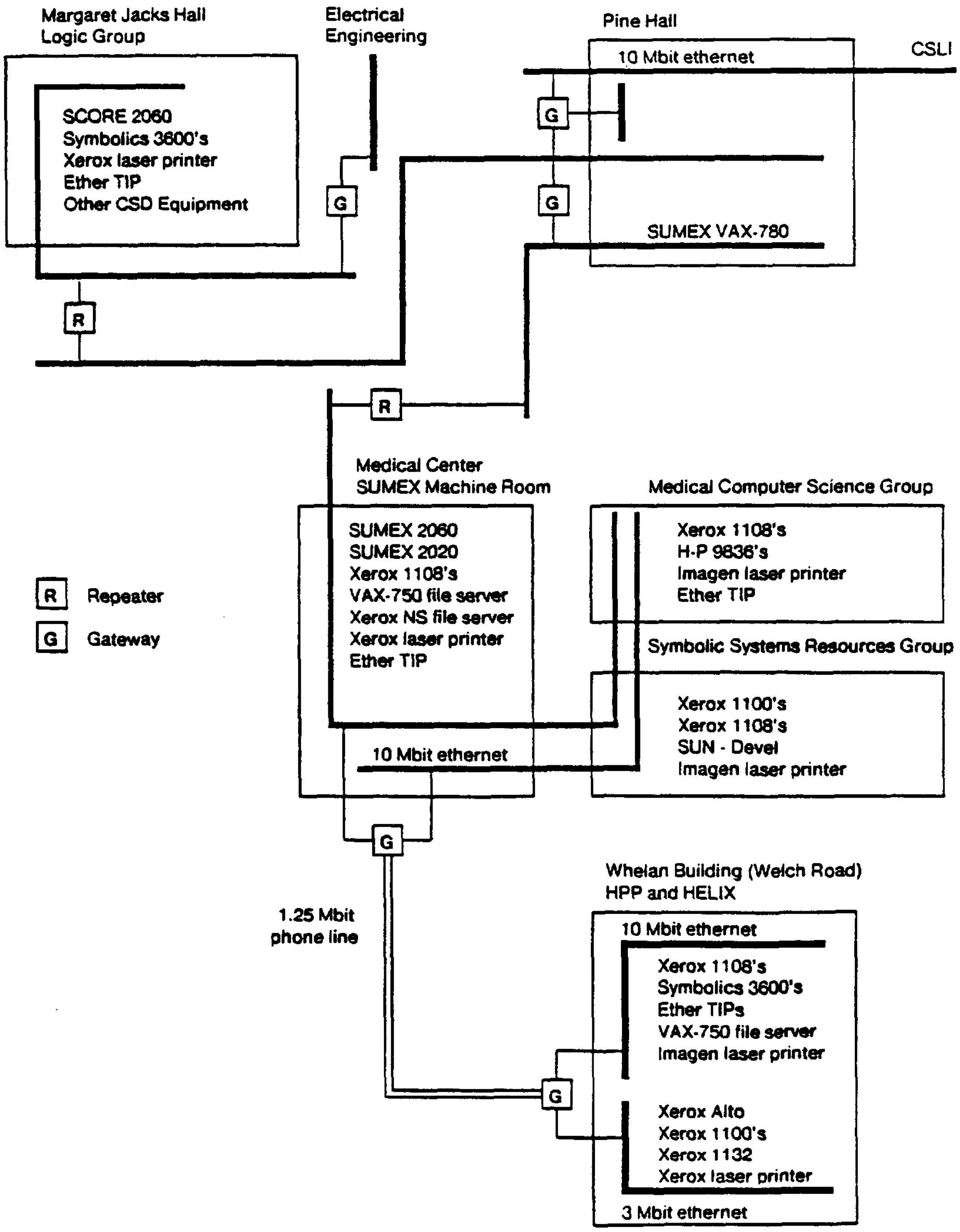

Figure 5: SUMEX-AIM EtherNet Configuration

lines up to 32 lines in a six slot backplane. With the advent of the newer 16 line DUART's developed in the Stanford Computer Science Department, 80 line TIP's have been built using this TIP code. This code is still running on several 3 Mbit/sec Ether TIP's at SUMEX. As 10 Mbit/sec networks were introduced, the TIP code was updated and adapted so that TIP's could run on either 3 MBit/sec or 10 MBit/sec Ethernets. There are now over 20 TIP's installed at Stanford using the SUMEX code and the number will increase substantially as the campus-wide local area network grows. The development of this software is essentially complete now with the recent addition of an improved user interface and facilities for inbound connections (such as for remote printers).

Ethernet Gateways -- Like the TIP systems, the PDP-11 gateway code was adapted to the MC-68000 hardware and extended to both 3 Mbit/sec and 10 Mbit/sec networks. Gateways can be configured to support up to four directly connected networks which may be either 3 MBit/sec or 10 MBit/sec. The gateway system was made "self-configuring" so that only one bootable gateway was needed. Network directory downloading and name/address lookup services were added. The routing algorithm was rewritten to minimize probe time for efficiency because of the continued growth of the number of subnetworks in the Stanford University network. The gateway now supports PUP and IP packet transport and XNS packet routing for both 10mb and 3mb networks is being completed. There are over twenty SUMEX gateways installed at Stanford and this number should double in the next year.

A special gateway configuration was required for the HPP move to Welch Road. Since the physical link was differentially driven 1.25 MBit/sec twisted pair cable, the network connections required two three-way gateways, one at either end, and special hardware to interface the serial lines with the ethernet interfaces. The required special hardware and software were built and the WR gateway has operated very effectively.

Apple Gateway Another special gateway, named SEAGATE, was developed to better integrate the Apple Macintosh into our Ethernet system. It links the Ethernet and Apple's AppleBus/AppleTalk network. This was completed and released in February 1985. Several internet sites, including some at Stanford, are currently constructing duplicate gateways. Also, several commercial firms are building a one board version of the gateway which should lower the cost to about $1000 per gateway. EFS, MAT, and AppleTalk Library are some sample Macintosh programs and UNIX daemons, that utilize SEAGATE. EFS is an external file system, written by John Seamons, and modified by us to work over AppleTalk. With EFS the Mac user sees his normal iconic view of the world. His UNIX directory appears as an icon and he can remotely execute and transfer files, simply by clicking on their icons. EFS is to the Mac as Leaf is to a LISP machine. The AppleTalk library is used by all of these programs to perform the ATP protocol (AppleTalk transaction protocol). This is the general protocol used to perform printing, file transfer, etc. with the Mac. The library allows a UNIX user-level process to perform this ATP protocol. Note that no kernel changes are required, since the ATP datagrams are imbedded in IP datagrams (UDP) by the SEAGATE. MAT is the Mac ATP Transfer program, a sample program that does file transfers with a UNIX host. It can also act as the framework for a Mac mail or print service.

Remote File Service -- In a distributed workstation environment, effective file access and transfer facilities between workstations and other hosts and servers are a must, especially to file servers like those we built around VAX 11/750 UNIX systems. Initial file service support used code written as a student project in the Stanford Computer Systems Laboratory. But as the number of workstations increased, service degraded and it became necessary to rewrite the PUP/BSP UNIX software package, and major portions of those programs dependent upon these protocols. This resulted in a 300% increase in throughput and stabilized the Lisp Machine to VAX 11/750 file service environment. At the same time we made major improvements to the UNIX Leaf service for XEROX D-machines. The earlier code, again a student systems project, had many bugs and inefficiencies and required a complete rewrite. In the new code, each Leaf connection was given a separate process to manage its Leaf resources, whereas previously, all users' Leaf requests were simply handled as a serial queue. This meant that every packet created a bottleneck for its successors. This work resulted in a much better Leaf service environment with considerable improvement in overall responsiveness and throughput.

Laser Printing Services

Since the first Xerox laser printers were developed in the mid-1970's, several companies have produced computer-driven systems, such as the Xerox Raven and the Imagen 8/300. These systems have become essential components of the work of the SUMEX-AIM community with applications ranging from scientific publications to hardcopy graphics output for ONCOCIN chemotherapy protocol patient charts. We have done much systems work to integrate laser printers into the SUMEX network environment so they would be routinely accessible from hosts and workstations alike.

We collaborated to develop an Ethernet interface for Imagen printers starting about January of 1984. We arranged to upgrade our Imprint-10 controller in exchange for the UNIX software needed to drive it from the network and were the first site to receive this controller in beta test stage. The UNIX software we developed made it possible to connect the printer to the new 4.2 BSD line printer spooler package using IP/TCP protocols. This was completed about March of 1984. After the UNIX implementation was complete, we developed the corresponding TOPS20 software to interface to this new printer and later, integrated it into the TOPS20 Galaxy spooler package. Other sites on campus and in the internet, began using the new printer and our spooling software as well.

We similarly developed and enhance the spooling system for the Dover and Alto-Raven laser printers and added a header page for Raven output to separate listings. And in addition to the device support for the printers to interface to the various mainframe hosts machines in our network, we also developed packages to allow Xerox D-machines and Symbolics 3600 machines to print to the networked laser printers.

On the SUMEX-AIM mainframe hosts, SCRIBE is the predominant document compilation system, but in the initial stages, it was essentially only used with the Xerox Dover printer or a daisywheel typewriter. In the succeeding years we integrated the Imagen Imprint-10 driver from Unilogic, brought up the Xerox Alto-Raven, and installed support for the new group of Imagen printers (the 8/300's), which are based on a Canon copier and are now the workhorse printing resources of the local community. We made numerous improvements in the printing fonts available to users, including a rework of Knuth's Computer Modern Roman fonts for a more contemporary look on the Imprint-10, creating a sans serif font family based on Computer Modern Roman, generating Helvetica and Times Roman font families from the Xerox sources used to generate the Dover fonts, and creating and improving many document types in use by the community.

General User Software

We have continued to assemble (develop where necessary) and maintain a broad range of user support software. These include such tools as language systems, statistics packages, DEC-supplied programs, text editors, text search programs, file space management programs, graphics support, a batch program execution monitor, text formatting and justification assistance, magnetic tape conversion aids, and user information/help assistance programs.

A particularly important area of user software for our community effort is a set of tools for inter-user communications. We have built up a group of programs to facilitate many aspects of communications including interpersonal electronic mail, a "bulletin board" system for various special interest groups to bridge the gap between private mail and formal system documents, and tools for terminal connections and file transfers between SUMEX and various external hosts. Examples of work on these sorts of programs have already been mentioned in earlier sections on operating systems and networking. A further gratifying example is the TTYFTP program, originally written at SUMEX as a system for file transfers usable over any circuit that appears as a terminal line to the operating system (hardline, dial-up, TYMNET, etc.) and incorporating appropriate control protocols and error checking. The design was derived from the DIALNET protocols developed at the Stanford AI Laboratory with extensions to allow both user and server modules to run as user processes without operating system changes. TTYFTP formed the basis for the KERMIT program that is now distributed by Columbia University and which is in very wide use for communications between personal computers and to mainframe hosts.

At SUMEX-AIM we are committed to importing rather than reinventing software where possible. As noted above, a number of the packages we have brought up are from outside groups. Many avenues exist for sharing between the system staff, various user projects, other facilities, and vendors. The availability of fast and convenient communication facilities coupling communities of computer facilities has made possible effective intergroup cooperation and decentralized maintenance of software packages. The many operating system and system software interest groups (e.g., TOPS-20, UNIX, D-Machines, network protocols, etc.) that have grown up by means of the ARPANET have been a good model for this kind of exchange. The other major advantage is that as a by-product of the constant communication about particular software, personal connections between staff members of the various sites develop. These connections serve to pass general information about software tools and to encourage the exchange of ideas among the sites and even vendors as appropriate to our research mission. We continue to import significant amounts of system software from other ARPANET sites, reciprocating with our own local developments. Interactions have included mutual backup support, experience with various hardware configurations, experience with new types of computers and operating systems, designs for local networks, operating system enhancements, utility or language software, and user project collaborations. We have assisted groups that have interacted with SUMEX user projects get access to software available in our community (for more details, see the section on Dissemination on page 81).

Operations and Support

The diverse computing environment that SUMEX-AIM provides requires a significant effort at operations and support to keep the resource responsive to community project needs. This includes the planning and management of physical facilities such as machine rooms and communications, system operations routine to backup and retrieve user files in a timely manner, and user support for communications, systems, and software advice. Of course, the upgrade of the KI-TENEX system to the 2060 required major planning and care to ensure continuous resource operation during the phase-over. Similarly, the relocation of our VAX 11/780 to Pine Hall and the outfitting of the KSL machine room at the Welch Road laboratory required much effort.

We use students for much of our operations and related systems programming work. Over the past 4 years, we have hired and trained a total of 15 undergraduate operations assistants.

We also spend significant time on new product review and evaluation such as Lisp workstations, terminals, communications equipment, network equipment, microprocessor systems, mainframe developments, and peripheral equipment. We also pay close attention to available video production and projection equipment, which has proved so useful in our dissemination efforts involving video tapes of our work.

III.A.3.4. Core AI Research

We have maintained a strong core AI research effort in the SUMEX-AIM resource aimed at developing information resources, basic AI research, and tools of general interest to the SUMEX-AIM community. It should be noted that the SUMEX resource grant from NIH provides much of the computing environment for this core AI work[1] but NIH supports only a small part of the manpower and other support for core AI. For example, NIH has provided partial funding for work on the AI Handbook, the AGE project, and part of the core ONCOCIN development for the dissemination of consultative AI systems. Substantial additional support for the personnel costs of our core AI research (roughly comparable to the NIH investment in computing resources) comes from DARPA, ONR, NSF, NASA, and several industrial basic research contracts to the Knowledge Systems Laboratory or KSL[2] (see the summary of core research funding on page 47).

Our core AI research work has long been the mainstay on which our extensive list of applications projects are based. This work has been focused on medical and biological problems for over a decade with considerable success, particularly in the area of expert systems which represent one important class of applications of AI to complex problems -- in medicine, science, engineering, and elsewhere. Numerous high-performance, expert systems have resulted from our work on expert systems in such diverse fields as analytical chemistry, medical diagnosis, cancer chemotherapy management, VLSI design, machine fault diagnosis, and molecular biology. Other projects have developed generalized software tools for representing and utilizing knowledge (e.g., EMYCIN [4, 34], UNITS [33], AGE [25], MRS [9], GLISP [27]) as well as comprehensive publications such as the three-volume *Handbook of Artificial Intelligence* [1] and books summarizing lessons learned in the DENDRAL [21] and MYCIN [4, 32] research projects.

But the current ideas fall short in many ways, necessitating extensive further basic research efforts. Our core research goals are to analyze the limitations of current techniques and to investigate the nature of methods for overcoming them. Long-term success of computer-based aids in medicine and biology depend on improving the programming methods available for representing and using domain knowledge.

The following summary reports progress on the basic or core research activities within the KSL. As indicated earlier, the development of the ONCOCIN system (under Professor Shortliffe) is an important part of our core research proposal for the renewal period. Progress on that work is reported separately in Section IV.A.3 on page 102, however, because its efforts have been supported as a collaborative and resource-related research project up until now. Together, this work explores a broad range of basic research ideas in many application settings, all of which contributes in the long term to improved knowledge based systems in biomedicine.

Recent Highlights of Research Progress

Research has progressed on several fundamental issues of AI. As in the past, our research methodology is experimental; we believe it is most fruitful at this stage of AI research to raise questions, examine issues, and test hypotheses in the context of specific problems such as management of patients with Hodgkins disease. Thus, within the KSL

---

[1] DARPA funds have also helped substantially in upgrading the KI-TENEX system to the 2060 and in the purchase of community Lisp workstations

[2] See Appendix A on page 203 for an overview of the KSL organization.

we build systems that implement our ideas for answering (or shedding some light on) fundamental questions; we experiment with those systems to determine the strengths and limits of the ideas; we redesign and test more; we attempt to generalize the ideas from the domain of implementation to other domains; and we publish details of the experiments. Many of these specific problem domains are medical or biological. In this way we believe the KSL has made substantial contributions to core research problems of interest not just to the AIM community but to AI in general.

In addition to the technical reports listed later, the following books and survey articles were published just during this year -- 11 books total have been published in the past 4 years as indicated in Appendix A. These are of central interest to AI researchers and of direct relevance to the mission of the SUMEX-AIM resource.

BOOKS:

1. Buchanan, B.G. and Shortliffe, E.H., eds. *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project.* Reading, MA: Addison-Wesley Publishing Company, 1984.
2. Clancey, W.J. and Shortliffe, E.H., eds. *Readings in Medical Artificial Intelligence: The First Decade.* Reading, MA: Addison-Wesley Publishing Company, 1984.
3. Cohen, Paul R. *Heuristic Reasoning about Uncertainty: An Artificial Intelligence Approach.* London and Marshfield, MA: Pitman Advanced Publishing Program, 1985.

SURVEY ARTICLES: HPP 84-15, 84-20, 84-23, 84-28, and 84-32.

In addition, work is progressing on a textbook for students beginning to study medical computing and artificial intelligence[1]. This multi-authored volume should be completed in draft form by the end of 1985 and a 1986 publication date is contemplated. Writing this new book will be facilitated by the SUMEX resource, much as the *Handbook of AI* was in the past. A multi-authored text of this type, particularly one for which the authors are spread at numerous different universities around the country, would be a nightmare to compile if it were not for the SUMEX resource. Many of the contributors to the book have been assigned SUMEX accounts for purposes of manuscript preparation. On-line manuscript work through the shared facility, coupled with messaging capabilities, will greatly enhance the efficiency and accuracy of the developing chapters and the editing process.

Progress is reported below under each of the major topics of our work. Citations are to KSL technical reports listed in the publications section.

1. *Knowledge representation*: How can the knowledge necessary for complex problem solving be represented for its most effective use in automatic inference processes? Often, the knowledge obtained from experts is heuristic knowledge, gained from many years of experience. How can this knowledge, with its inherent vagueness and uncertainty, be represented and applied?

   A working version of NEOMYCIN has been implemented which demonstrates the effectiveness of representing strategy knowledge explicitly. A detailed study of rule-based systems was published in book form. Specific representational issues in logic-based systems were addressed in the

[1]Shortliffe, E.H., Wiederhold, G.C.M., and Fagan, L.M.; *An Introduction to Medical Computer Science.* Reading, MA: Addison-Wesley (in preparation).

context of MRS. We designed a method for representing temporal knowledge in ONCOCIN. Finally, Cooper's Ph.D. thesis on representing and using causal and probabilistic knowledge was published in this year.

[See KSL technical memos KSL-84-9, KSL-84-10, KSL-84-18, KSL 84-31, KSL-84-41, KSL-85-5.]

2. *Advanced Architectures and Control*: What kinds of software tools and system architectures can be constructed to make it easier to implement expert programs with increasing complexity and high performance? How can we design flexible control structures for powerful problem solving programs?

   Much of our research in the past year has involved investigations with the Blackboard architecture begun in previous years. We have implemented our design in a working system called BB1.

   [See KSL technical memos KSL-84-11, KSL-84-12, KSL-84-14, KSL 84-16, KSL 84-36.]

3. *Knowledge Acquisition*: How is knowledge acquired most efficiently -- from human experts, from observed data, from experience, and from discovery? How can a program discover inconsistencies and incompleteness in its knowledge base? How can the knowledge base be augmented without perturbing the established knowledge base?

   Three Ph.D. theses (Fu, Greiner, and Dietterich) in the area of knowledge acquisition were completed in this year. Fu's work develops methods for learning by induction, where the target rules may have some associated degrees of uncertainty and may contain names of intermediate concepts. This work was demonstrated in the context of diagnosing causes of jaundice. Greiner's work examines learning by analogy. Dietterich's work elucidates methods needed in learning programs to deal with state variables and with problems of using a partially learned theory to interpret new data that will be used to learn new elements of the theory. In addition, we implemented the first parts of a program that can learn by watching an expert. And we implemented a prototype system that learns control heuristics from an expert using a problem solving program written in BB1.

   [Preliminary results have been published in KSL-84-10, KSL-84-18, KSL-84-24, KSL-84-38, KSL-84-45, KSL 84-46, KSL-85-2, KSL-85-4.]

4. *Knowledge Utilization*: By what inference methods can many sources of knowledge of diverse types be made to contribute jointly and efficiently toward solutions? How can knowledge be used intelligently, especially in systems with large knowledge bases, so that it is applied in an appropriate manner at the appropriate time?

   We completed the design of a system using Dempster's rule of propagating uncertainty, and we examined several other issues regarding the use of probabilistic information in expert systems. Dr. Jean Gordon, a mathematician and Stanford medical student, collaborated with Dr. Shortliffe on work that examines inexact inference using the Dempster-Shafer theory of evidence, demonstrating its relevance to a familiar expert system domain, namely the bacterial organism identification problem that lies at the heart of the MYCIN system, and presenting a new adaptation of the D-S approach with both computational efficiency and permitting the management of evidential reasoning within an abstraction hierarchy.

   We examined the use of counter-factual conditionals in logic-based systems and completed an analysis of how procedural hints can be used by a problem solver.

[See KSL technical memos KSL-84-11, KSL-84-17, KSL-84-21, KSL-84-30, KSL-84-31, KSL-84-35, KSL 84-41, KSL-84-42, KSL-84-42, KSL-84-43.]

5. *Software Tools*: How can specific programs that solve specific problems be generalized to more widely useful tools to aid in the development of other programs of the same class?

   We have continued the development of new software tools for expert system construction and the distribution of packages that are reliable enough and documented so that other laboratories can use them. These include the old rule-based EMYCIN system, MRS, and AGE. Progress has been made in making the BB1 instantiation of the blackboard architecture domain-independent. We have begun constructing and editing subsystems and have completed a first implementation of an explanation subsystem.

   [See KSL technical memos KSL-84-16, KSL-84-39.]

6. *Explanation and Tutoring*: How can the knowledge base and the line of reasoning used in solving a particular problem be explained to users? What constitutes a sufficient or an acceptable explanation for different classes of users? How can knowledge in a system be transferred effectively to students and trainees?

   A program for inferring a model of users was designed and implemented in the context of a tutoring system that aids in teaching algebra. A second user-modelling program was implemented in the context of NEOMYCIN to help understand how an expert solves problems. A survey of explanation capabilities in medical consultation programs was published.

   A new project on knowledge-based explanations in a decision analysis environment is getting underway as the thesis research of Dr. Glenn Rennels. This work is actually a synthesis of artificial intelligence, decision analysis and statistics. The work concerns medical management, not diagnosis; diagnostic decisions identify underlying mechanisms of the illness, and group the patient's problems under a diagnostic label, whereas management decisions plan actions that will prevent undesirable outcomes and restore health. The intelligent behavior we want to emulate is (a) the identification of studies relevant to a given clinical case, and (b) interpretation of those studies for decision-making assistance.

   [See KSL technical memos KSL-84-12, KSL 84-27, KSL-84-29.]

7. *Planning and Design*: What are reasonable and effective methods for planning and design? How can symbolic knowledge be coupled with numerical constraints? How are constraints propagated in design problems?

   A major paper on skeletal planning was published in this year. And we published in the biochemistry literature some results of applying skeletal planning to experiment design in genetic engineering.

   [See KSL technical memos KSL-84-33, KSL-85-6.]

8. *Diagnosis*: How can we build a diagnostic system that reflects any of several diagnostic strategies? How can we use knowledge at different levels of abstraction in the diagnostic process?

   Research on using causal models in a medical decision support system (NESTOR) was published in this year. Using the domain of hypercalcemic disorders, NESTOR attempts to use knowledge-based methods within a formal probability theory framework. The system is able to score hypotheses with causal knowledge guiding the application of sparse probabilistic knowledge; search for the most likely hypothesis without exploring the entire hypothesis space; and critique and compare hypotheses which are generated by the system, volunteered by the user, or both.

A second medical diagnosis program that uses causal models of renal physiology (AI/MM) was also published. In this system, analysis and explanation of physiological function is based on two kinds of causal relations: empirical "Type-1" relations based on definitions or on repeated observation and mathematical "Type-2" relations that have a basis in physical law. Inference rules are proposed for making valid qualitative causal arguments with both kinds of causal basis.

A working implementation of the PATHFINDER system was evaluated and its diagnostic strategies were analyzed. A taxonomy of diagnostic methods was completed and integrated into the NEOMYCIN system.

[See KSL technical reports: KSL-84-13, KSL-84-19, KSL-84-48, KSL-85-5.]

Relevant Core Research Publications

HPP 84-9 David H. Hickam, Edward H. Shortliffe, Miriam B. Bischoff, A. Carlisle Scott, and Charlotte D. Jacobs; *Evaluations of the ONCOCIN System: A Computer-Based Treatment Consultant for Clinical Oncology, (1) The Quality of Computer-Generated Advice and (2) Improvements in the Quality of Data Management,* May 1984.

HPP 84-10 Thomas G. Dietterich; *Learning About Systems That Contain State Variables,* June 1984. In *Proceedings of AAAI-84,* August 1984.

HPP 84-11 M. Genesereth, and D.E. Smith; *Procedural Hints in the Control of Reasoning,* May 1984.

HPP 84-12 Derek H. Sleeman; *UMFE: A User Modelling Front End Subsystem,* April 1984.

HPP 84-13 Eric J. Horvitz, David E. Heckerman, Bharat N. Nathwani, and Lawrence M. Fagan; *Diagnostic Strategies in the Hypothesis-Directed PATHFINDER System,* June 1984, submitted to the *First Conference on Artificial Intelligence Applications, Denver, CO.,* December 5-7, 1984.

HPP 84-14 Vineet Singh, and M. Genesereth; *A Variable Supply Model for Distributing Deductions,* May 1984.

HPP 84-15 Bruce G. Buchanan; *Expert Systems,* July 1984, *Journal of Automated Reasoning, Vol. 1, No. 1, Fall,* 1984.

HPP 84-16 *STAN-CS-84-1034* Barbara Hayes-Roth; *BB-1: An Architecture for Blackboard Systems That Control, Explain, and Learn About Their Own Behavior,* December 1984.

HPP 84-17 M.L. Ginsberg; *Analyzing Incomplete Information,* 1984.

HPP 84-18 William J. Clancey; *Knowledge Acquisition for Classification Expert Systems,* July 1984, *Proceedings of ACM-84,* 1984.

HPP 84-19 E.H. Shortliffe; *Coming to Terms With the Computer,* to appear in S.R. Reiser, and M. Anbar (eds.), *The Machine at the Bedside: Strategies for Using Technology in Patient Care, Cambridge University Press,* 1984.

HPP 84-20 E.H. Shortliffe; *Artificial Intelligence and the Future of Medical Computing*, in *Proceedings of a Symposium on Computers in Medicine, annual meeting of the California Medical Association, Anaheim, CA.*, February 1984.

HPP 84-21 E.H. Shortliffe; *Reasoning Methods in Medical Consultation Systems: Artificial Intelligence Approaches (Tutorial)*, in *Computer Programs in Biomedicine* January 1984.

HPP 84-22 *ONCOCIN Project: Studies to Evaluate the ONCOCIN System; 6 Abstracts*, February 1984.

HPP 84-23 Edward H. Shortliffe; *Feature Interview: On the MYCIN Expert System*, in *Computer Compacts, 1:283-289*, December 1983/January 1984.

HPP 84-24 B.G. Buchanan, and E.H. Shortliffe; *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*, published with *Addison-Wesley, Reading, MA.*, 1984.

HPP 84-25 W.J. Clancey, and E.H. Shortliffe; *Readings in Medical Artificial Intelligence: The First Decade*, published with *Addison-Wesley, Reading, MA.*, 1984.

HPP 84-27 Edward H. Shortliffe; *Explanation Capabilities for Medical Consultation Systems (Tutorial)*, in D. Lindberg, and M. Collen (eds.), *Proceedings of AAMSI Congress 84, pp. 193-197, San Francisco*, May 21-23, 1984.

HPP 84-28 E.H. Shortliffe, and L.M. Fagan; *Artificial Intelligence: The Expert Systems Approach to Medical Consultation*, in *Proceedings of the 6th Annual International Symposium on Computers in Critical Care and Pulmonary Medicine, Heidelberg, Germany*, June 4-7, 1984.

HPP 84-29 David C. Wilkins, Bruce G. Buchanan, and William J. Clancey: *Inferring an Expert's Reasoning by Watching, Proceedings of the 1984 Conference on Intelligent Systems and Machines*, 1984.

HPP 84-30 M.L. Ginsberg: *Non-Monotonic Reasoning Using Dempster's Rule*, June 1984.

HPP 84-31 M.L. Ginsberg: *Implementing Probabilistic Reasoning*, June 1984.

HPP 84-32 Bruce G. Buchanan: *Artificial Intelligence: Toward Machines That Think*, July 1984, in *Yearbook of Science and the Future, pp. 96-112, Encyclopedia Britannica, Inc., Chicago*, 1985.

HPP 84-33 Rene Bach, Yumi Iwasaki, and Peter Friedland; *Intelligent Computational Assistance for Experiment Design*, in *Nuclear Acids Research*, January 1984.

MCS Thesis Kunz, John C.; *Use of Artificial Intelligence and Simple Mathematics to Analyze a Physiological Model*, Doctoral dissertation, Medical Information Sciences, June 1984.

HPP 84-35 Jean Gordon, and Edward Shortliffe; *A Method for Managing Evidential Reasoning in a Hierarchical Hypothesis Space*, September 1984 and in *Artificial Intelligence*, 26(3), July 1985.

HPP 84-36 Michael R. Genesereth, Matt Ginsberg, and Jeff S. Rosenschein; *Cooperation Without Communication*, September 1984.

HPP 84-38	Li-Min Fu, and Bruce G. Buchanan; *Enhancing Performance of Expert Systems by Automated Discovery of Meta-Rules,* September 6, 1984.
HPP 84-39	Paul S. Rosenbloom, John E. Laird, John McDermott, Allen Newell, and Edmund Orciuch; *R1-Soar: An Experiment in Knowledge-Intensive Programming in a Problem-Solving Architecture,* to appear in the *Proceedings of the IEEE Workshop on Principles of Knowledge-Based Systems,* October 1984.
HPP 84-41	*STAN-CS-84-1032* Michael R. Genesereth, Matthew L. Ginsberg, and Jeffrey S. Rosenschein; *Solving the Prisoner's Dilemma,* November 1984.
HPP 84-42	Matthew L. Ginsberg; *Does Probability Have a Place in Non-Monotonic Reasoning?* submitted to the *IJCAI-85,* November 1984.
HPP 84-43	*STAN-CS-84-1029* Matthew L. Ginsberg; *Counterfactuals,* submitted to the *IJCAI-85,* December 1984.
HPP 84-45	Devika Subramanian, and Michael R. Genesereth; *Experiment Generation with Version Spaces,* December 1984.
HPP 84-46	Thomas G. Dietterich; *Constraint Propagation Techniques for Theory-Driven Data Interpretation,* PhD Thesis, to be published as a book by *Kluwer,* December 1984.
HPP 84-48	*STAN-CS-84-1031* Gregory F. Cooper; *NESTOR: A Computer-Based Medical Diagnostic Aid That Integrates Causal and Probabilistic Knowledge,* PhD Thesis, December 20, 1984.
KSL 85-2	*STAN-CS-85-1036* Barbara Hayes-Roth, and Michael Hewett; *Learning Control Heuristics in BB1,* submitted to the *IJCAI-85,* January 1985.
KSL 85-4	(Needs Authors Permission) Li-Min Fu, and Bruce G. Buchanan; *Learning Intermediate Knowledge in Constructing a Hierarchical Knowledge Base,* submitted to the *IJCAI Conference Proceedings for 1985,* January 1985.
KSL 85-5	(Needs Authors Permission) William J. Clancey; *Heuristic Classification,* March 1985.
KSL 85-6	Peter E. Friedland, and Yumi Iwasaki; *The Concept and Implementation of Skeletal Plans,* published in the *Journal of Automated Reasoning,* 1985.
KSL 85-7	Rene Bach, Yumi Iwasaki, and Peter Friedland; *Intelligent Computational Assistance for Experiment Design,* published in *Nucleic Acids Research,* 1985.
KSL 85-8	(Needs Authors Permission) M.G. Kahn, J. Ferguson, E.H. Shortliffe, and L. Fagan; *An Approach for Structuring Temporal Information in the ONCOCIN System,* March 1985.

Summary of Core Research Funding Support

We are pursuing a broad core research program on basic AI research issues with support from not only SUMEX but also DARPA, NASA, NSF, and ONR. SUMEX provides some salary support for staff and students involved in core research and invaluable computing support for most of these efforts.

Interactions with the SUMEX-AIM Resource

Our interactions with the SUMEX-AIM resource involve the facilities -- both hardware and software -- and the staff -- both technical and administrative. Taken together as a whole resource, they constitute an essential part of the research structure for the KSL. Many of the grants and contracts from other agencies have been awarded partly because of the cost-effectiveness of AI research in the KSL due to the fact that much of our computing needs could be more than adequately met by the SUMEX-AIM resource. In this way the complementary funding of this work by the NIH and other agencies provides a high leverage for incremental investment in AI research at the SUMEX-AIM resource.

We rely on the central SUMEX facility as a focal point for all the research within the KSL, not only for much of our computing, but for communications and links to our many collaborators as well. As a common communications medium alone, it has significantly enhanced the nature of our work and the reach of our collaborations. The existence of the central time-shared facility has allowed us to explore new ideas at very small incremental cost.

As SUMEX and the KSL acquire a diversity of hardware, including LISP workstations and smaller personal computers, we rely more and more heavily on the SUMEX staff for integration of these new resources into the local network system. The staff has been extremely helpful and effective in dealing with the myriad of complex technical issues and leading us competently into this world of decentralized, diversified computing. At the same time, the staff has provided a stable, efficient central time-shared machine running software that has been developed at many sites over many years. Without the dedication of the SUMEX staff, the KSL would not be at the forefront of AI research.

III.A.3.5. Training Activities

The SUMEX resource exists to facilitate biomedical artificial intelligence applications from program development through testing in the target research communities. This user orientation on the part of the facility and staff has been a unique feature of our resource and is responsible in large part for our success in community building. The resource staff has spent significant effort in assisting users gain access to the system and use it effectively. We have also spent substantial effort to develop, maintain, and facilitate access to documentation and interactive help facilities. The HELP and Bulletin Board subsystems have been important in this effort to help users get familiar with the computing environment.

On another front, we have regularly accepted a number of scientific visitors for periods of several months to a year, to work with us to learn the techniques of expert system definition and building and to collaborate with us on specific projects. Our ability to accommodate such visitors is severely limited by space, computing, and manpower resources to support such visitors within the demands of our on-going research.

And finally, the training of graduate students is an essential part of the research and educational activities of the KSL. Currently 41 students are working with our projects centered in Computer Science and another 20 students are working with the Medical Computer Science program in Medicine. Of the 41 working in Computer Science, 25 are working toward Ph.D. degrees, and 16 are working toward M.S. degrees. A number of students are pursuing interdisciplinary programs and come from the Departments of Engineering, Mathematics, Education, and Medicine.

Based on the SUMEX-AIM community environment, we have initiated two unique and special academic degree programs at Stanford, the Medical Information Science program and the Masters of Science in AI, to increase the number of students we produce for research and industry, who are knowledgeable about knowledge-based system techniques.

The *Medical Information Sciences (MIS)* program is one of the most obvious signs of the local academic impact of the SUMEX-AIM resource. The MIS program received recent University approval (in October 1982) as an innovative training program that offers MS and PhD degrees to individuals with a career commitment to applying computers and decision sciences in the field of medicine. The MIS training program is based in School of Medicine, directed by Dr. Shortliffe, co-directed by Dr. Fagan, and overseen by a group of nine University faculty that includes several faculty from the Knowledge Systems Laboratory (Profs. Shortliffe, Feigenbaum, Buchanan, and Genesereth). It was Stanford's active ongoing research in medical computer science, plus a world-wide reputation for the excellence and rigor of those research efforts, that persuaded the University that the field warranted a new academic degree program in the area. A group of faculty from the medical school and the computer science department argued that research in medical computing has historically been constrained by a lack of talented individuals who have a solid footing in both the medical and computer science fields. The specialized curriculum offered by the new program is intended to overcome the limitations of previous training options. It focusses on the development of a new generation of researchers with a commitment to developing new knowledge about optimal methods for developing practical computer-based solutions to biomedical needs.

The program accepted its first class of four trainees in the summer of 1983 and a second class of five entered last summer. A third group of seven students has just been selected to begin during 1985. The proposed steady state size for the program (which should be reached in 1986) is 20-22 trainees. Applicants to the program in our first two years have come from a number of backgrounds (including seven MD's and five medical students). We do not wish to provide too narrow a definition of what kinds of prior training are pertinent because of the interdisciplinary nature of the field. The program has accordingly encouraged applications from any of the following:

- medical students who wish to combine MD training with formal degree work and research experience in MIS;
- physicians who wish to obtain formal MIS training after their MD or their residency, perhaps in conjunction with a clinical fellowship at Stanford Medical Center;
- recent BA or BS graduates who have decided on a career applying computer science in the medical world;
- current Stanford undergraduates who wish to extend their Stanford training an extra year in order to obtain a "co-terminus" MS in the MIS program;
- recent PhD graduates who wish post-doctoral training, perhaps with the formal MS credential, to complement their primary field of training.

In addition, a special one-year MS program is available for established academic medical researchers who may wish to augment their computing and statistical skills during a sabbatical break.

With the exception of this latter group, all students spend a minimum of two years at Stanford (four years for PhD students) and are expected to undertake significant research projects for either degree. Research opportunities abound, however, and they of course include the several Stanford AIM projects as well as research in psychological and formal statistical approaches to medical decision making, applied instrumentation, large medical databases, and a variety of other applications projects at the medical center and on the main campus. Several students are already contributing in major ways to the AIM projects and core research described in this application.

Early evidence suggests that the program already has an excellent reputation due to:

- high quality students, many of whom are beginning to publish their work in conference proceedings and refereed journals;
- a rigorous curriculum that includes newly-developed course offerings that are available to the University's medical students, undergraduates, and computer science students as well as to the program's trainees;
- excellent computing facilities combined with ample and diverse opportunities for medical computer science and medical decision science research;
- the program's great potential for a beneficial impact upon health care delivery in the highly technologic but cost-sensitive era that lies ahead.

The program has been successful in raising financial and equipment support (almost $1M in hardware gifts from Hewlett Packard, Xerox, and Texas Instruments; over $200K in cash donations from corporations and foundations; and an NIH post-doctoral training grant from the National Library of Medicine).

The *Master of Science in Computer Science: Artificial Intelligence (MS:AI)* program is a terminal professional degree offered for students who wish to develop a competence in the design of substantial knowledge-based AI applications but who do not intend to obtain a Ph.D. degree. The MS:AI program is administered by the Committee for Applied Artificial Intelligence, composed of faculty and research staff of the Computer Science Department. Normally, students spend two years in the program with their